

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1-15. (Canceled)

16. (Presently Amended) A system for extending functionality of a class object, comprising:

a processing unit;
a system memory in communication with the processing unit via a system bus;
a computer-readable medium in communication with the processing unit via the system bus; and
an extensible object model executed from the computer-readable medium by the processing unit, wherein the extensible object model determines whether a requested functionality is inherent in the class object, and wherein the extensible object model causes the processing unit to create an extension object from an existing extension package when a requested functionality is not inherent in the class object, and wherein the extension object extends the class object to provide the requested functionality.

17. (Previously presented) The computerized system of claim 16, wherein the extensible object model further causes the processing unit to notify the extension object when the extension is deleted.

18. (Original) The computerized system of claim 16, wherein the extensible object model further causes the processing unit to register the extension package in an extension database stored on the computer-readable medium.

19. (Original) The computerized system of claim 16, wherein the extensible object model further causes the processing unit to store the extension object in system memory when the corresponding extension is first referenced.

20. (Original) The computerized system of claim 16, wherein the extensible object model further causes the processing unit to create the extension object from the extension package

by causing the processing unit to create an extension provider object and causes the processing unit to create the extension from the extension provider object.

21. (Original) The computerized system of claim 16, wherein the extensible object model further causes the processing unit to create an event filtering and sourcing object to handle events generated by the extension object.

22-26. (Canceled)

27. (Presently Amended) A computer-readable medium having stored thereon computer-executable components comprising:

an extensible object;
an extension database having an entry for an extension for the extensible object; and

an existing extension package having an interface for obtaining an extension object that provides the extension for the extensible object, wherein the extensible package determines whether a requested functionality is inherent in the class object.

28. (Original) The computer-readable medium of claim 27, further comprising:

an extension provider object that proffers the extension object as a result of a call to the interface in the extension package.

29. (Presently Amended) A method for extending functionality of a class object in a run-time environment, comprising:

determining whether a requested functionality is inherent in the class object;
receiving a request from an application for functionality that is not inherent in the class object;

determining if the functionality is available in a first extension object;
obtaining an existing extension package having computer-executable instructions associated with the extension object functionality, wherein the extension package proffers an extension provider object when the functionality is requested;

specifying parameters to the extension provider object to create a second extension object; and

directing the request to the second extension object.

30. (Previously presented) The method of claim 29, further comprising registering the extension package in an extension database.

31. (Previously presented) The method of claim 29, further comprising storing the extension package in an extension database.

32. (Previously presented) The method of claim 31, further comprising searching for an entry associated with the functionality in the extension database to determine if the functionality is available in the extension object.

33. (Previously presented) The method of claim 29, further comprising creating the second extension object when the extended functionality is first referenced, and locating the second extension object when the extended functionality is subsequently referenced.

34. (Previously presented) A method for extending functionality of a class object in a run-time environment, comprising:

determining whether a requested functionality is inherent in the class object; receiving a request from an application for functionality that is not inherent in the class object;

determining if the functionality is available in a first extension object; and

directing the request to the functionality in a second extension object, when the functionality is not available in the first extension object.

35. (Previously presented) The method of claim 34, further comprising obtaining an extension package having computer-executable instructions associated with the extension object functionality.

36. (Previously presented) The method of claim 35, further comprising storing the extension package in an extension database.

37. (Previously presented) The method of claim 35, further comprising registering the extension package in an extension database.

38. (Previously presented) The method of claim 34, wherein the extension package proffers an extension provider object when the functionality is requested.

39. (Previously presented) The method of claim 38, wherein the extension provider object creates the extension object.

40. (Previously presented) The method of claim 34, further comprising searching for an entry associated with the functionality in an extension database.

41. (Previously presented) The method of claim 34, further comprising creating the second extension object when the extended functionality is first referenced, and locating the second extension object when the extended functionality is subsequently referenced.

42. (Presently Amended) A system for extending functionality of a class object, comprising:
a processing unit;
a system memory in communication with the processing unit via a system bus;
a computer-readable medium in communication with the processing unit via the system bus;
an extensible object model executed from the computer-readable medium by the processing unit, wherein the extensible object model determines whether a requested functionality is inherent in the class object, and wherein the extensible object model creates an extension object from an existing extension package when a requested functionality is not inherent in the class object, and wherein the extension object extends the class object to provide the requested functionality.

43. (Previously presented) The system of claim 42, wherein information about the extension package is stored in an extension database.

44. (Previously presented) A method for extending functionality of a class object, comprising:

determining whether a requested functionality is inherent in the class object;
invoking a functionality that is not inherent in the class object;
determining if the invoked functionality is available in a first extension object;
creating a second extension object when the invoked functionality is not available in the first extension object; and
directing the invocation to the second extension object.